



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **08171487 A**(43) Date of publication of application: **02 . 07 . 96**

(51) Int. Cl.

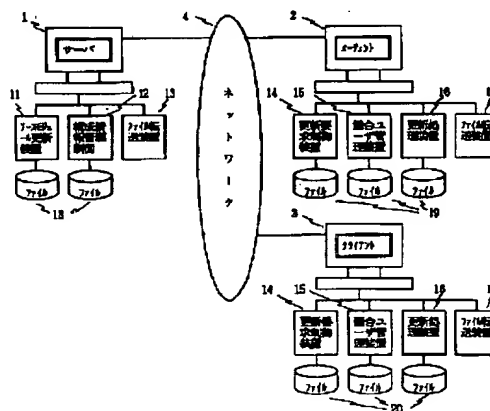
G06F 9/06
G06F 15/00(21) Application number: **06314748**(22) Date of filing: **19 . 12 . 94**(71) Applicant: **FUJITSU LTD**(72) Inventor: **IKUTA KATSUICHI**
AOYAMA MIKIO(54) **SOFTWARE UPDATE MANAGEMENT SYSTEM**

COPYRIGHT: (C)1996,JPO

(57) Abstract:

PURPOSE: To simultaneously update the same module by stepwise subdividing the module and performing contention control for respective subdivided components, to reduce a file transfer amount to be the cause of the congestion of a network and to shorten file transfer time by subdividing the module.

CONSTITUTION: A source module updating device 11 is an update management system provided with an update history management system and performs update history management to constitution information and components for constituting the module respectively. A constitution information management controller 12 judges whether or not the pertinent module is during updating at present at the time of a module update request from a client 3. If it is not during the updating, the controller 12 copies the constitution information on a base line module as dynamic constitution information and manages it along with the component information during the updating at present. At the time of the transfer request for the constitution information and components from the client 3, file transfer devices 13 and 17 perform file transfer between a server and an agent/the client connected through the network 4.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-171487

(43)公開日 平成8年(1996)7月2日

(51)Int.Cl.⁶

G 0 6 F 9/06
15/00

識別記号

5 4 0 A 7230-5B
3 1 0 U 9364-5L

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数6 O L (全 16 頁)

(21)出願番号 特願平6-314748

(22)出願日 平成6年(1994)12月19日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72)発明者 生田 勝一

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者 青山 幹雄

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

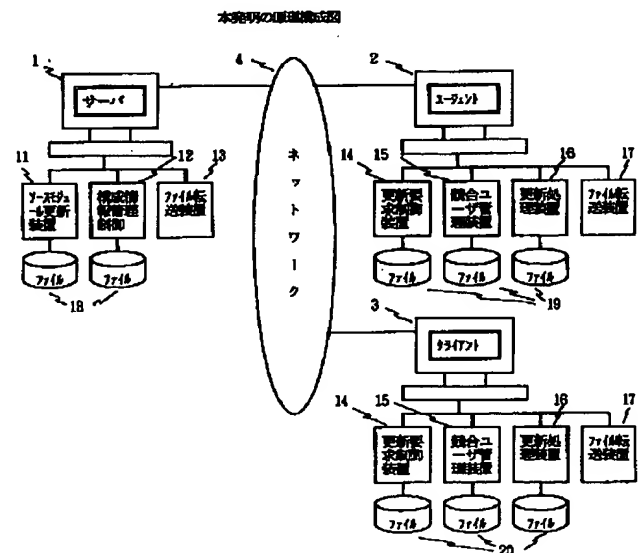
(74)代理人 弁理士 井桁 貞一

(54)【発明の名称】 ソフトウェア更新管理方式

(57)【要約】

【目的】 ネットワークを介して接続される分散開発環境下におけるソフトウェア更新管理方式に関し、モジュール内を細分化することにより、ファイル転送量の縮小とファイル転送時間の短縮を図ることを目的とする。

【構成】 サーバにて、構成情報及び部品の更新を行うモジュール更新装置と、部品単位に管理される更新者情報の制御を行う構成情報管理制御装置と、クライアントにて、更新する部品の更新要求を発行する更新要求制御装置と更新処理装置と、エージェントにて、更新する部品が競合した場合の制御を行う競合ユーザ管理装置とを備え、更新中エージェント以外の更新要求を競合ユーザ管理装置にて競合ユーザとして更新待ちユーザ情報を登録し、更新終了後に部品及び更新履歴をエージェントとクライアント間で転送するように構成する。



【特許請求の範囲】

【請求項 1】 分散開発環境下で、ソースモジュールを複数の部品に分割し、ソースモジュールを構成する部品の構成情報の変更管理と、部品の変更管理の 2 段階で管理するソフトウェアの構成管理方式において、サーバにて、構成情報及び部品の更新を行うモジュール更新装置と、部品単位に管理される更新者情報の制御を行う構成情報管理制御装置と、クライアントにて、更新する部品の更新要求を発行する更新要求制御装置と更新処理装置と、エージェントにて、更新する部品が競合した場合の制御を行う競合ユーザ管理装置とを備え、更新中エージェント以外の更新要求を競合ユーザ管理装置にて競合ユーザとして更新待ちユーザ情報を登録し、更新終了後に部品及び更新履歴をエージェントとクライアント間で転送する事により、ソースモジュールを格納するサーバへのアクセス回数の削減、及び部品単位でのファイル転送を行うことを特徴とするソフトウェア更新管理方式。

【請求項 2】 前記ソフトウェアの構成管理方式において、構成情報に部品単位の更新中表示を設け、他の更新者からの同時更新を禁止する装置、一つの部品を更新中に他の部品を同一画面上に展開する装置、更新中に競合する部品を更新するエージェントからの更新完了通知を受信する装置を備え、複数の部品を同時に更新する場合に、更新可能な部品の転送要求と、他のエージェントにて更新中の部品の更新待ち要求を発行し、更新中エージェントの更新完了通知受信時に、随時更新部品をクライアントにて機能する更新画面上に展開することを特徴とする請求項 1 記載のソフトウェア更新管理方式。

【請求項 3】 前記ソフトウェアの構成管理方式において、競合モジュールの更新要求を行う際に更新の予約状況を表示する装置を備え、エージェントにて管理される更新待ちユーザ情報に更新開始時刻、更新期間テーブルを付加する事により、更新待ちクライアントはサーバにて管理される更新中ユーザ情報を基に更新中ユーザの更新待ち情報を検索し、更新待ちクライアントが自更新時のスケジューリングを可能とすることを特徴とする請求項 1 記載のソフトウェア更新管理方式。

【請求項 4】 前記ソフトウェアの構成管理方式において、サーバにて管理されるダイナミック構成情報中に記述される更新部品毎のエージェント情報を基に、他の部品の更新要求が生じた場合に、更新中のエージェントの構成情報内の部品情報を、更新中に変更する更新中情報制御装置を備え、更新する部分のみを更新中表示にする事で、他の部品に

て変更が生じた事を、該当するエージェントへ通知することを特徴とする請求項 1 記載のソフトウェア更新管理方式。

【請求項 5】 前記ソフトウェアの構成管理方式において、更新済みクライアントが、自分の更新履歴を保管する機能を備え、更新中エージェントがサーバに対し復元要求を発行し、サーバはダイナミック構成情報中のエージェント情報を基に、更新済みクライアントから更新履歴情報を復元し、エージェントへ送付する事により、更新中のソースモジュールの復元を可能とすることを特徴とする請求項 1 記載のソフトウェア更新管理方式。

【請求項 6】 前記ソフトウェアの構成管理方式において、クライアントは更新待ち申請時のバックアップテーブルと、更新中のエージェントからの更新完了通知が更新申請時に登録する更新開始時間を越えていないかを監視する装置を設け、更新待ち申請時に登録した更新開始時間迄にエージェントからの更新完了通知が来なかった場合に、エージェントに対して更新期間が遅延している事を通知することを特徴とする請求項 1 記載のソフトウェア更新管理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はネットワークを介して接続される分散開発環境下におけるソフトウェア更新管理方式に関する。ソフトウェアの更新管理方式は、分散開発環境下ではソースモジュールを複数の部品に分割し、ソースモジュールを構成する部品の構成情報の変更管理と、部品の変更管理の 2 段階でサーバが管理する。

【0002】サーバは、ソースモジュールの更新が完了した場合に、ベースラインモジュールとして一元的に管理する装置であり、クライアントは、ネットワークを介してサーバに一元的に管理される端末装置群であり、クライアントにて更新するソースモジュールの更新が完了した場合に、ベースラインモジュールとしてサーバの格納装置に登録される。

【0003】

【従来の技術】従来のソフトウェア開発は、ホストによる一括集中型のソースモジュール管理を行い、他のユーザよりの更新を禁止することにより競合制御を行っている。従来の一括集中型のソースモジュール管理方法を図 1 2 に示す。

【0004】図において、21はホスト装置、22はネットワーク、23、24は端末装置を示す。ホスト装置21はメインフレームであり、データソースを保有する。ネットワーク22はホスト装置21と複数の端末装置23、24とを接続し、端末装置23のモジュールの変更情報によりホスト装置21のソースモジュールを更新し、その間、他の端末装置24からのモジュール変更は受け付けない。

【0005】

【発明が解決しようとする課題】分散開発環境下を中心とするソフトウェアの開発形態により、1モジュール当たりのソース規模の増大、ファイル転送時間の増大、及びファイル転送に使用するネットワークの輻輳等の問題が発生している。

【0006】開発量の増大による競合モジュール増加の解決手段として、競合モジュールをスケジューリングする専任者を設けることにより問題を解決していたが、モジュール規模の増大によるソフト更新工数の増大、開発周期の短縮化に伴い短時間内でのスケジューリングによる対応が困難となっている。

【0007】本発明は、モジュール内を段階的に細分化し、細分化した部品毎に競合制御を行うことにより、同一モジュールの一斉更新を可能とすると同時に、モジュールを細分化することにより、ネットワークの輻輳の原因となるファイル転送量の縮小、及びファイル転送時間の短縮を図ることを目的とする。

【0008】

【課題を解決するための手段】本発明の原理構成図を図1に示す。図において、1はサーバ、2はエージェント、3はクライアント、4はネットワーク、11はソースモジュール更新装置、12は構成情報管理制御装置、13はファイル転送装置、14は更新要求制御装置、15は競合ユーザ管理装置、16は更新処理装置、17はファイル転送装置、18、19、20はサーバ、エージェント、クライアントのファイル群を示す。

【0009】サーバ1は、ソースモジュールの更新が完了した場合に、ベースラインモジュールとして一元的に管理する装置である。エージェント2とクライアント3は端末装置であり、エージェント2はサーバ代行端末装置である。

【0010】ソースモジュール更新装置11はクライアントで更新するソースモジュールにて更新が完了した場合に、ベースラインモジュールとしてサーバの格納装置へ登録する装置であり、構成情報管理制御装置12はクライアントより転送される更新部品情報を格納する装置であり、ファイル転送装置13は更新部品、構成情報等をエージェント、クライアントに転送する装置である。

【0011】更新要求制御装置14はクライアントにて更新する部品を指定する装置で、競合ユーザ管理装置15はクライアントからの更新待ち要求情報を競合ユーザ管理テーブルに登録する装置であり、更新処理装置16は構成情報上に展開された部品を更新する装置であり、ファイル転送装置17は更新した差分情報、競合ユーザ管理テーブルをサーバ又はクライアントに転送する装置である。

【0012】なお、エージェント2とクライアント3の競合ユーザ管理装置15に接続されるファイル19、20は競合ユーザ管理テーブルを示し、ネットワークを介してサーバ又はクライアント、エージェント間で転送される。

【0013】

【作用】分散開発環境下で、ソースモジュールを複数の部品に分割し、ソースモジュールを構成する部品の構成情報の変更管理と、部品の変更管理の2段階で管理するソフトウェアの構成管理方式において、サーバ1にて、構成情報及び部品の更新を行うモジュール更新装置11と、部品単位に管理される更新者情報の制御を行う構成情報管理制御装置12と、クライアント3にて、更新する部品の更新要求を発行する更新要求制御装置14と、更新処理装置16と、エージェント2にて、更新する部品が競合した場合の制御を行う競合ユーザ管理装置15とを備え、更新中エージェント2以外の更新要求を競合ユーザ管理装置15にて競合ユーザとして更新待ちユーザ情報を登録し、更新終了後に部品及び更新履歴をエージェント2とクライアント3間で転送する事により、ソースモジュールを格納するサーバ1へのアクセス回数の削減、及び部品単位でのファイル転送を行うことでファイル転送時間の短縮を図ることができる。

【0014】

【実施例】本発明の実施例1のシステム構成図を図2に示す。図において、図1と同一番号は同一装置名を示す。サーバ1、エージェント2、クライアント3の各装置に接続されるファイル群18、19、20は、それぞれ次のように構成される。

【0015】サーバ1のソースモジュール更新装置11には、モジュール一覧、構成情報、その更新履歴、部品S M、その更新履歴の各ファイルを有し、構成情報管理制御装置12はダイナミック構成情報管理テーブルを有する。ダイナミック構成情報管理テーブルにはモジュールの状態、部品名、使用者名が登録されている。

【0016】エージェント2の更新要求制御装置14は構成情報管理テーブルを有し、競合ユーザ管理装置15は競合ユーザ管理テーブルを有し、競合ユーザ管理テーブルには部品名と更新待ちクライアント名が登録されている。更新処理装置16は更新履歴バックアップファイルを有する。また、クライアント3の更新要求制御装置14は更新予約バックアップファイルを有する。

【0017】ソースモジュール更新装置11は、更新履歴管理システムを有する更新管理システムであり、モジュールを構成する構成情報、部品のそれぞれに更新履歴管理を行う装置である。構成情報管理制御装置12は、クライアント3からのモジュール更新要求時に該当するモジュールが現在更新中か判断し、更新中でなければベースラインモジュールの構成情報をダイナミック構成情報として複写し、現在更新中の部品情報と共に管理する装置である。

【0018】クライアント3からの構成情報、部品の転送要求時に、ファイル転送装置13、17は、ネットワーク4を介して接続されるサーバとエージェント/クライアント間のファイル転送を行う装置である。更新要求制御

装置14は、構成情報の更新状況とモジュール間の構造を示す図、例えば木構造ルーチンエディタにて表示し、更新箇所の指定をサーバ/エージェントに対して送るものである。

【0019】競合ユーザ管理装置15は、更新情報をサーバ1より受信した時にエージェント2としての機能を行う装置であり、更新待ち要求がクライアント3より発行された場合に、更新待ち情報として競合ユーザ管理テーブルに更新ユーザ情報を登録する。更新処理装置16は構成情報、部品を更新する為の木構造チャート等の更新エディタを備える。

【0020】本発明の実施例1の処理フローチャートを図3と図4に示す。図はクライアントAとクライアントBとサーバとの間の処理フローを示す。各装置とも操作とデータと処理との関連を示す。図において、

(1) クライアントAは更新するモジュールの更新情報をサーバより抽出する為の要求を更新要求制御装置より発行する。

(2) サーバの構成情報管理装置は、更新要求モジュールのダイナミック構成情報が存在するか判定し、なければベースライン構成情報をダイナミック構成情報に複写し、要求を発行したクライアントAにダイナミック構成情報を送信する。

(3) 構成情報を受信したクライアントAは、自分の更新箇所が更新中でないことを確認し、構造チャートエディタにて更新箇所にマークし、ファイル転送装置にてサーバに更新部品転送要求を発行する。

(4) サーバは構成情報テーブルにユーザ情報としてユーザのネットワークアドレスを付加し、ダイナミック構成情報管理テーブルを改版した上で、指定される部品ファイルをクライアントAに転送する。

(5) 更新する部品を受信したクライアントAは、木構造チャートエディタにて構成情報をオープンし、指定した部品部分にサーバより転送された部品の内容を展開し、更新処理を行う。

(6) クライアントAをエージェントAとし、エージェントAにて更新中に、他のクライアントBにて更新中部品の更新要求が発生した場合、サーバより抽出したダイナミック構成情報ファイル上の更新する部分の更新者情報より、ネットワークアドレスを抽出し、ファイル転送装置にて更新中のエージェントAに対し、更新予約の要求を発行する。

(7) 更新中のエージェントAは更新予約情報として予約者のネットワークアドレスを更新待ち管理テーブルへ追加する。

(8) エージェントAは更新処理が完了した場合、更新待ちファイルに更新待ち情報が存在するか検索し、

(9) 更新待ち情報が存在しなければファイル転送装置にて更新履歴をサーバに転送して、ソースモジュール登録を終了する。

(10) 更新予約が存在した場合、更新待ちクライアントBに対し、自更新処理が終了したことを通知する。

(11) 更新終了通知を受信したクライアントBは、更新情報受信が可能であるというオーダをエージェントAに通知する。

(12) オーダを受信したエージェントAは、更新情報として更新履歴、更新した部品、更新予約管理テーブルをクライアントBに送付する。

【0021】以上の手順により、サーバにて管理されるソースモジュール中の更新要部品のみの更新およびファイル転送を行うことができる。本発明の実施例2の処理フローチャートを図5と図6に示す。図は、クライアントAが概要記述を更新中に、クライアントBが概要記述部と変数宣言の2つの部品を更新する場合を示す。

【0022】更新要求制御装置にて更新部品の登録を行う際に、ある部品にて更新中だった場合に、ある部品の更新要求と更新中エージェントに対し、更新待ち要求を発行する機能を設け、更新処理装置にてソースモジュールを更新中に、更新待ち部品の更新完了通知の受信処理を行う場合、競合ユーザ制御装置を並行に起動し、更新完了通知が受信された場合に、更新中の木構造チャートへ随時、更新部品を展開する。図において、

(21) クライアントAが概要記述を更新中に、

(22) クライアントBの更新要求制御装置は、競合部分の部品について、更新中のエージェントAに対し更新待ち要求を発行する。

(23) 更新が可能な部品は、サーバに対して部品の転送要求を発行し、更新可能な部品の更新作業を行う。

(24) 競合する部品を更新中のエージェントAからの更新完了通知が更新待ちのクライアントBに発行された場合、更新待ち部品の完了通知を、更新処理と並行して起動される競合エージェントAに対し、更新部品の送信要求を発行し、自端末に更新する部品を取り込む。

(25) 更新可能部分のみを更新中だった更新処理装置は、競合部分の更新が可能になる事を更新中の更新者に通知し、部品を展開するか否かを確認し、木構造チャートエディタ上に部品の詳細情報を展開し、更新可能状態とする。

【0023】本発明の実施例3のシステム構成図を図7に示す。図において、図2と同一番号は同一装置名を示し、サーバ1とエージェント2とクライアント3とをネットワーク4を介して接続し、エージェント2は競合ユーザ管理テーブルに、部品名毎に更新待ちクライアント名、更新開始日、更新期間を登録し、これを更新するように構成する。クライアント3はUSRNとして更新要求制御装置に構成要求バックアップファイルを有する。

【0024】部品更新中のエージェント2は更新する情報として、競合ユーザ管理装置に更新スケジュールとして更新期間を登録する。他のクライアント3にて同一部品の競合が発生した場合、更新待ちクライアント3は更

新中エージェント 2 に対し、更新スケジュールの転送要求を発行する。

【0025】更新中エージェント 2 は更新要求を受信後、更新待ちクライアント 3 に競合ユーザ管理テーブルを送信する。更新待ちクライアント 3 は競合ユーザ管理テーブルを参照し、自更新のスケジュールリングを行い、更新中エージェント 2 に対し更新待ち要求を発行する。

【0026】本発明の実施例 4 の処理フローチャートを図 8 に示す。クライアント A とクライアント B とサーバとの間の処理フローを示す。

(31) クライアント A により、サーバにて管理される構成情報を取り込み、他の部品にて更新が行われている場合に、構成情報内の部品使用クライアント情報により、更新中クライアント B に対して更新表示依頼を発行する。

(32) 更新中のエージェント A は、他の部品での更新が発生した事を更新中のユーザに通知する為に、更新中の木構造チャートエディタ上の更新部分の更新状態を使用中にする。

【0027】本発明の実施例 5 の処理フローチャートを図 9 と図 10 に示す。実施例 5 は、更新処理装置にて抽出される更新履歴、及び競合ユーザ管理装置に格納される自更新者情報を、サーバからの更新履歴収集要求時に更新履歴を次更新者に送付する。図において、

(41) 複数のエージェントを介して更新作業を行うクライアント B にて、更新途中に何らかの要因にて更新不能となった場合に、更新中のエージェント B からサーバに対し、更新前状態への復旧要求を発行する。

(42) サーバはダイナミック構成情報ファイル中の復旧部品の先頭エージェントを検索し、先頭エージェント A に対し更新履歴収集要求を発信する。

(43) 収集要求を受信したエージェント A は更新情報バックアップテーブルに格納される更新履歴を抽出し、競合ユーザ管理テーブルに管理される次更新者に対し更新情報収集要求を発行する。

(44) 復旧要求を発行したクライアント A は復旧要求が次端末に発行されたことを確認し、更新履歴をマージする事により更新前状態に更新部品を復旧する事が可能となる。

(45) 更新作業が完了し、サーバへ更新情報が転送された時点にて、サーバは構成情報テーブルの先頭更新者 A に対し、モジュールの登録が終了した事を通知し、バックアップ情報を削除する。

【0028】本発明の実施例 6 の処理フローチャートを図 11 に示す。実施例 6 は、更新要求制御装置より発行した自更新スケジュールを保管し、自更新開始日にエージェントからの更新完了通知が転送されることをチェックし、更新完了通知が転送されない場合にエージェントに対しし更新状態の報告要求を発行する。

【0029】図 11 の処理フローチャートにおいて、

(51) 更新待ちクライアント B にて更新待ち要求を発行し

た時の情報をバックアップし、更新待ちクライアント B は更新開始時間迄に更新完了通知が転送されることをチェックし、

(52) 更新完了通知が更新中エージェント A より送付されない場合は、更新エージェント A に対して更新期間を越えている事を更新要求制御装置より通知する。

【0030】

【発明の効果】

①. 実施例 1 により、ソースモジュールを細分化し、細分化した部品単独のファイル転送、競合制御を複数のエージェントにより独立して行い、サーバの輻輳を軽減し、同時にネットワークの輻輳を軽減する効果がある。

②. 実施例 2 により、複数の部品を更新する場合にて、更新する部品の一部が使用中でも、更新中のエージェントによる更新が終了した時点で更新部分を自更新端末に展開でき、複数回の更新作業をまとめて更新できる。

③. 実施例 3 により、更新の予約時、空き時間が存在すれば自由な時間に更新する事が可能となる。

④. 実施例 4 により、更新中モジュール中の他の部品に、他のクライアントにて更新処理が発生した事を通知すると同時に、エージェントにて保管される構成情報と同一レベルにする事により、サーバ上に管理される構成情報を再度取り込まなくても、更新中の構成情報を基に更新部品の追加を行うことが可能となる。

⑤. ソフトウェア開発時に、不慮の電源の切断、更新情報のセーブ忘れ等により更新情報が破壊される場合が多々存在する。実施例 5 により、更新中の不慮のデータ破壊時、自更新部分のみの再更新にて更新情報の復旧を行う事ができる。

⑥. 実施例 6 により、更新遅延者に更新期間の遅延通知を行うことにより、更新忘れの予防、更新スケジュールの厳守を遂行することができる。

【図面の簡単な説明】

【図 1】 本発明の原理構成図

【図 2】 実施例 1 のシステム構成図

【図 3】 実施例 1 の処理フローチャート (その 1)

【図 4】 実施例 1 の処理フローチャート (その 2)

【図 5】 実施例 2 の処理フローチャート (その 1)

【図 6】 実施例 2 の処理フローチャート (その 2)

【図 7】 実施例 3 のシステム構成図

【図 8】 実施例 4 の処理フローチャート

【図 9】 実施例 5 の処理フローチャート (その 1)

【図 10】 実施例 5 の処理フローチャート (その 2)

【図 11】 実施例 6 の処理フローチャート

【図 12】 従来の一括集中型のソースモジュール管理方法

【符号の説明】

1 サーバ

2 エージェント

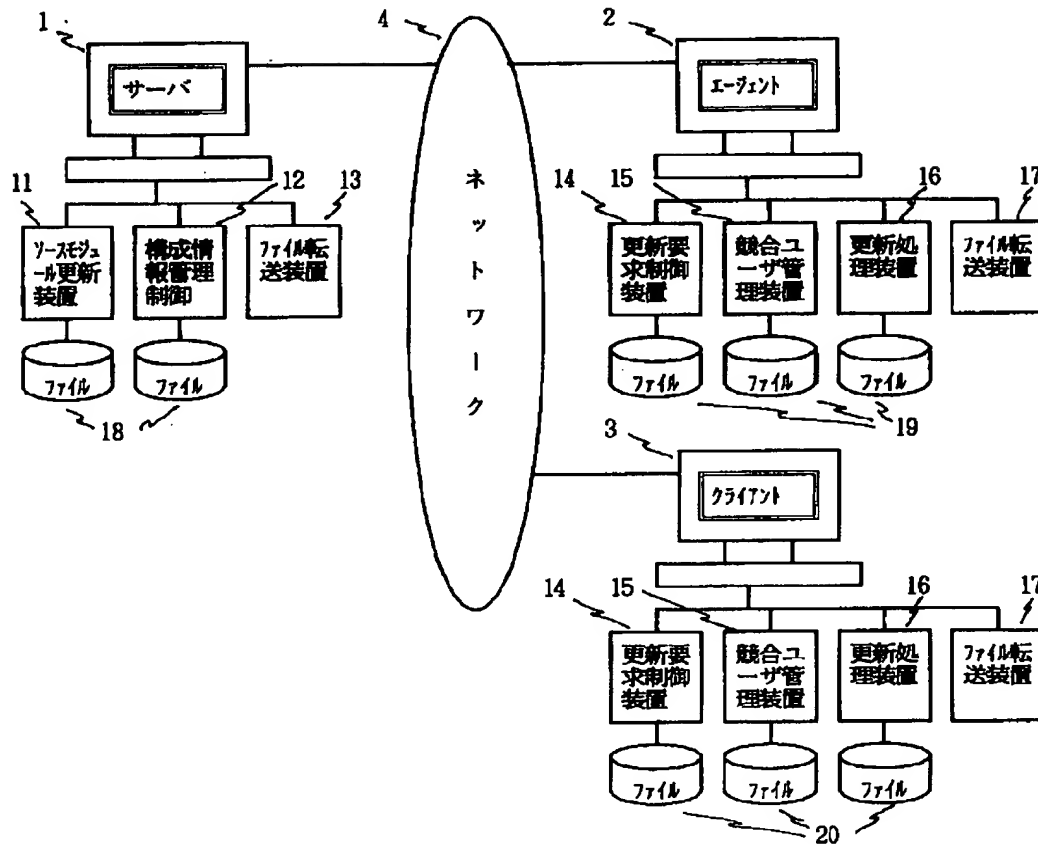
3 クライアント

- 4 ネットワーク
- 11 ソースモジュール更新装置
- 12 構成情報管理制御装置
- 13, 17 ファイル転送装置
- 14 更新要求制御装置
- 15 競合ユーザ管理装置

- * 16 更新処理装置
- 17 ファイル転送装置
- 18, 19, 20 ファイル装置
- 21 ホスト装置
- 22 ネットワーク
- * 23, 24 端末装置

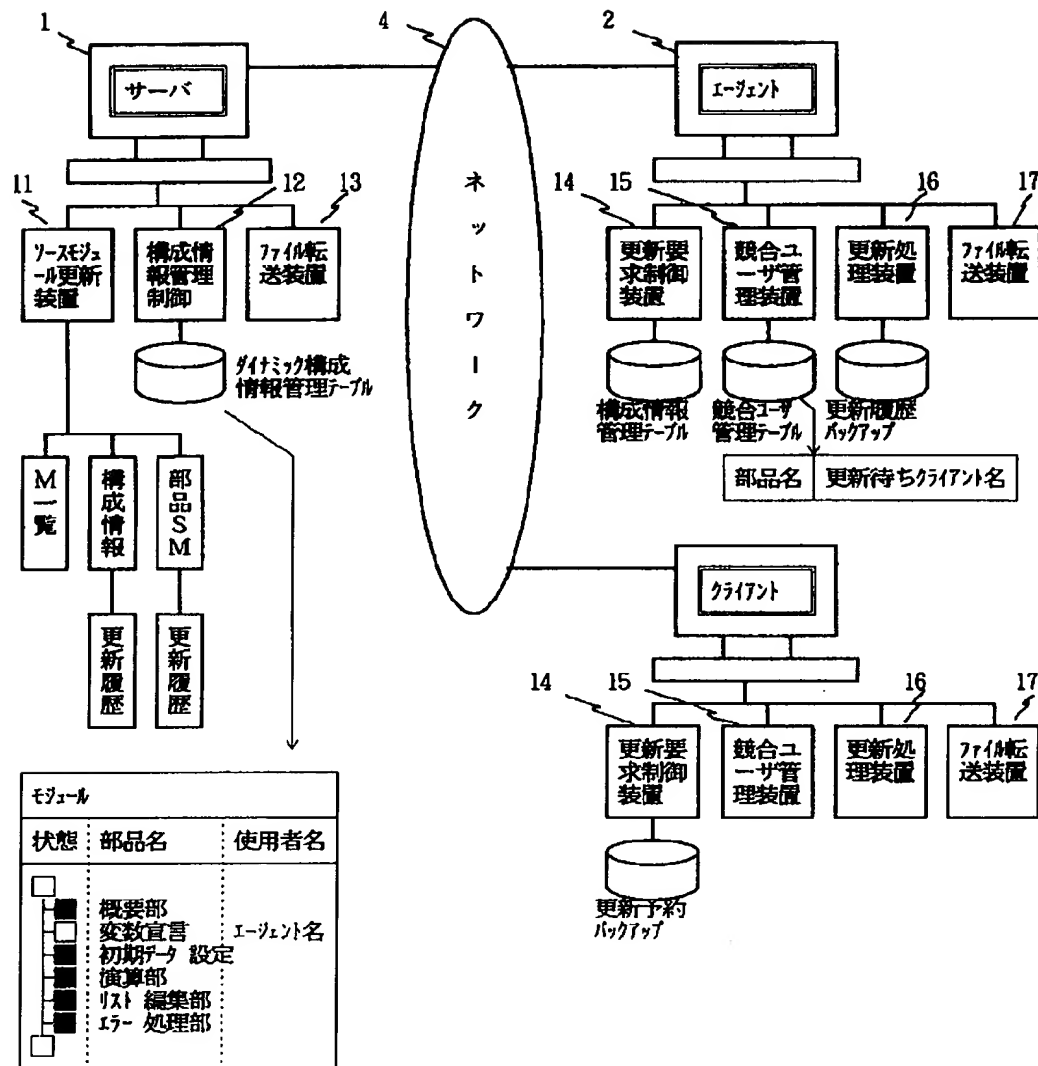
【図1】

本発明の原理構成図



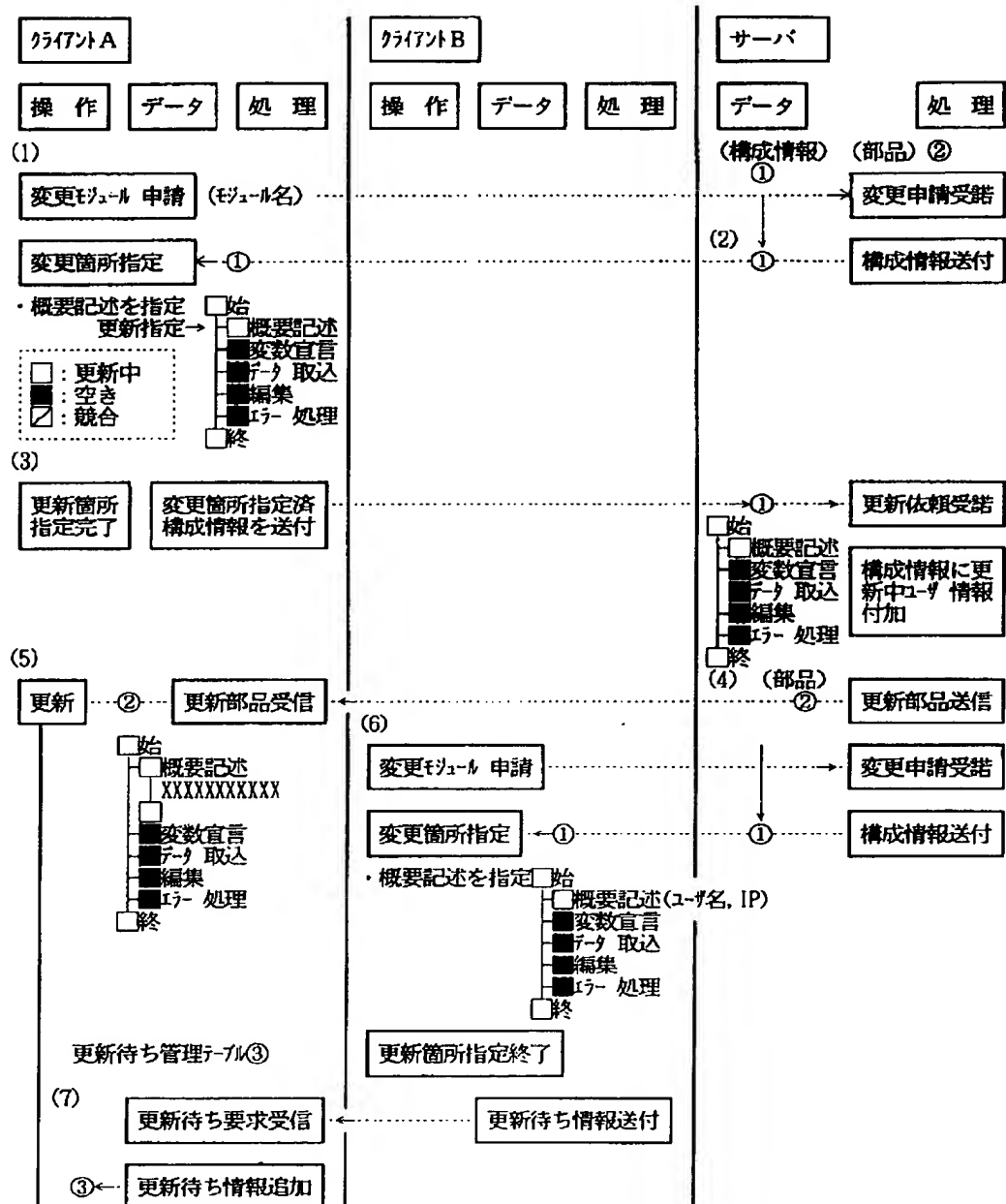
【図 2】

実施例 1 のシステム構成図



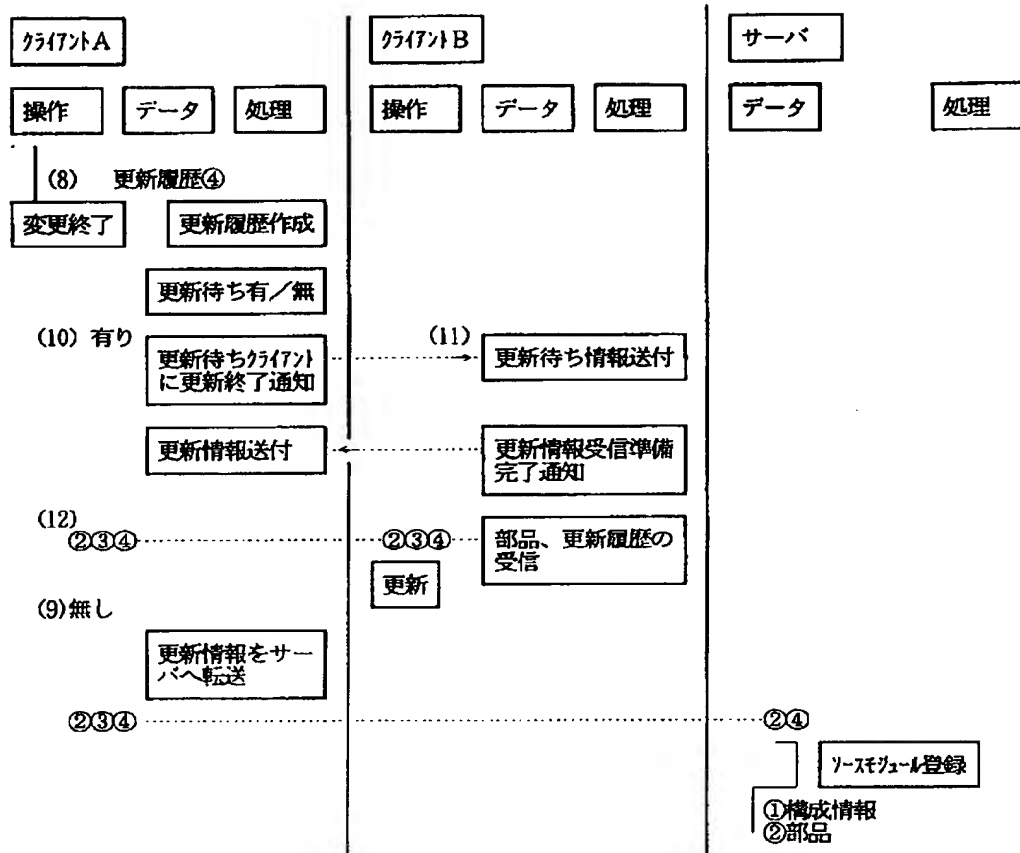
【図 3】

実施例 1 の処理フローチャート (その 1)



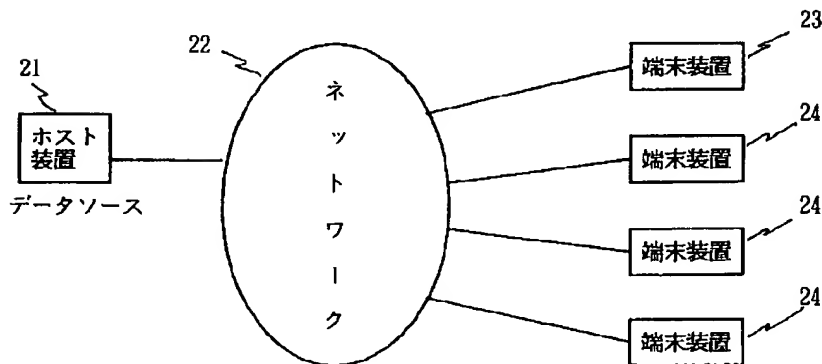
【図 4】

実施例 1 の処理フローチャート (その 2)



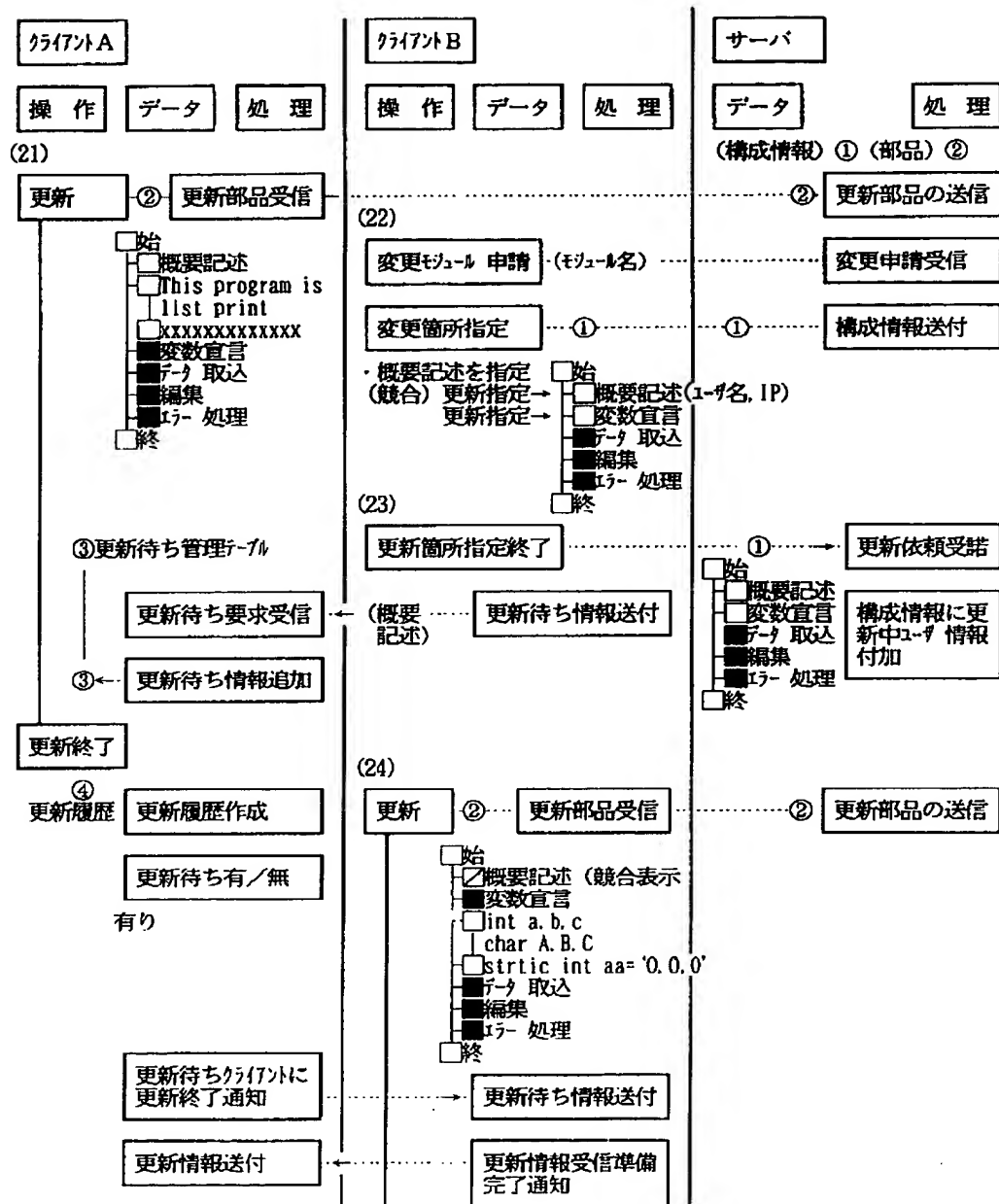
【図 1 2】

従来の一括集中型のソースモジュール管理方法



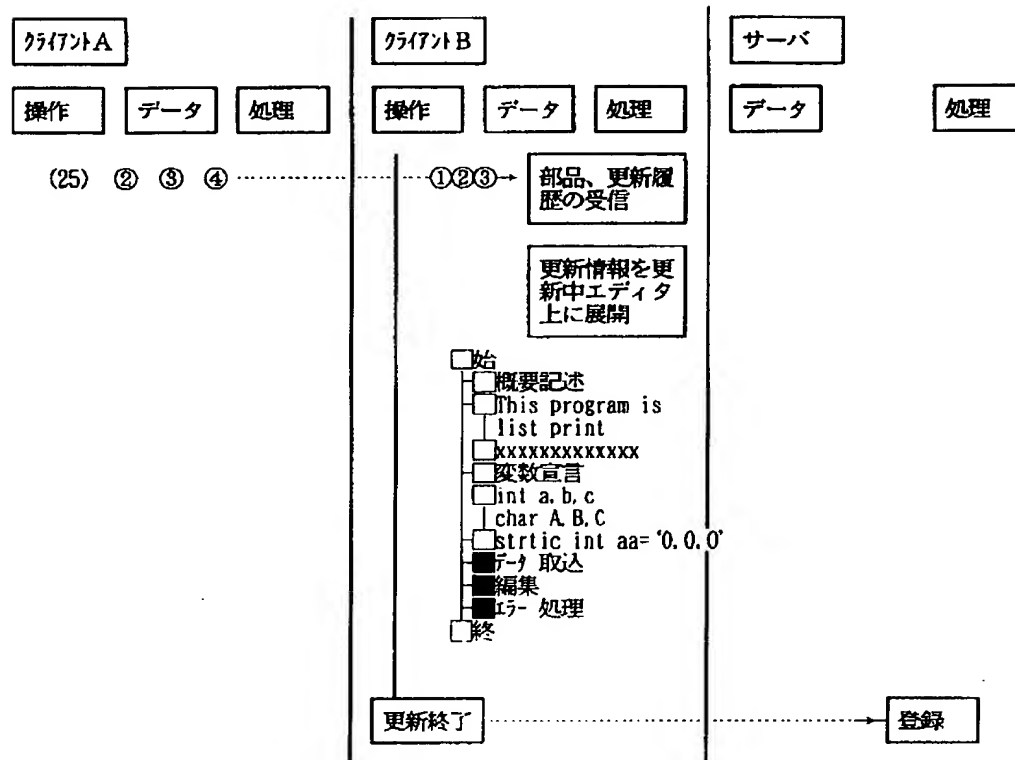
【図 5】

実施例 2 の処理フローチャート (その 1)



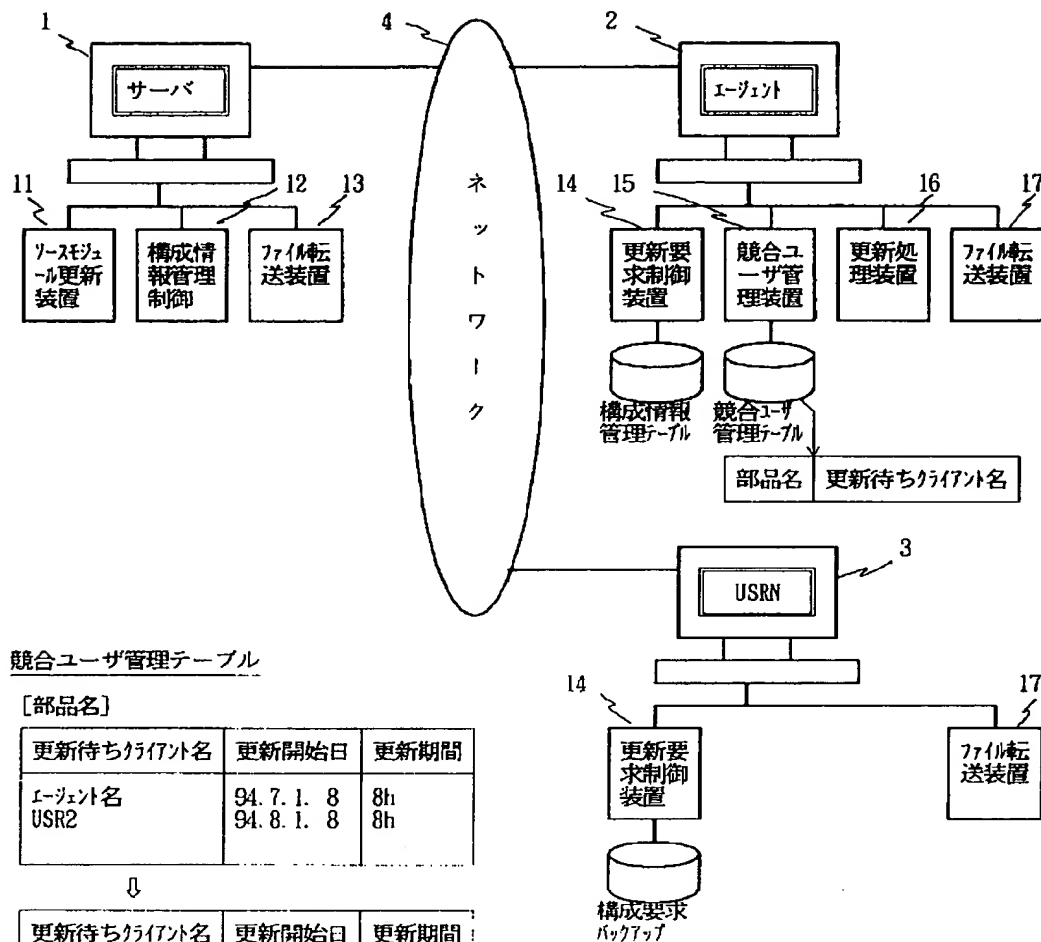
【図6】

実施例2の処理フローチャート（その2）



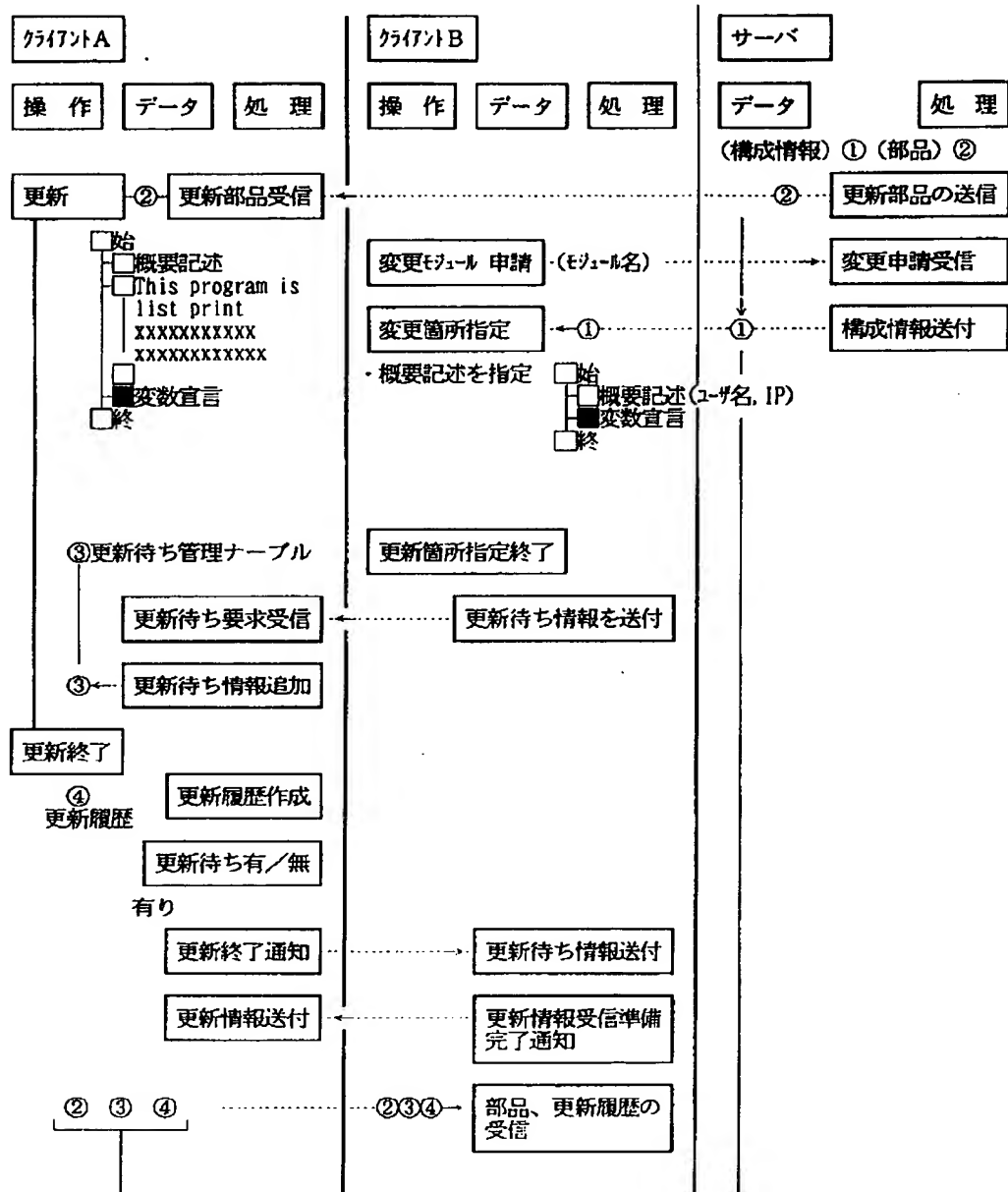
【図 7】

実施例 3 のシステム構成図



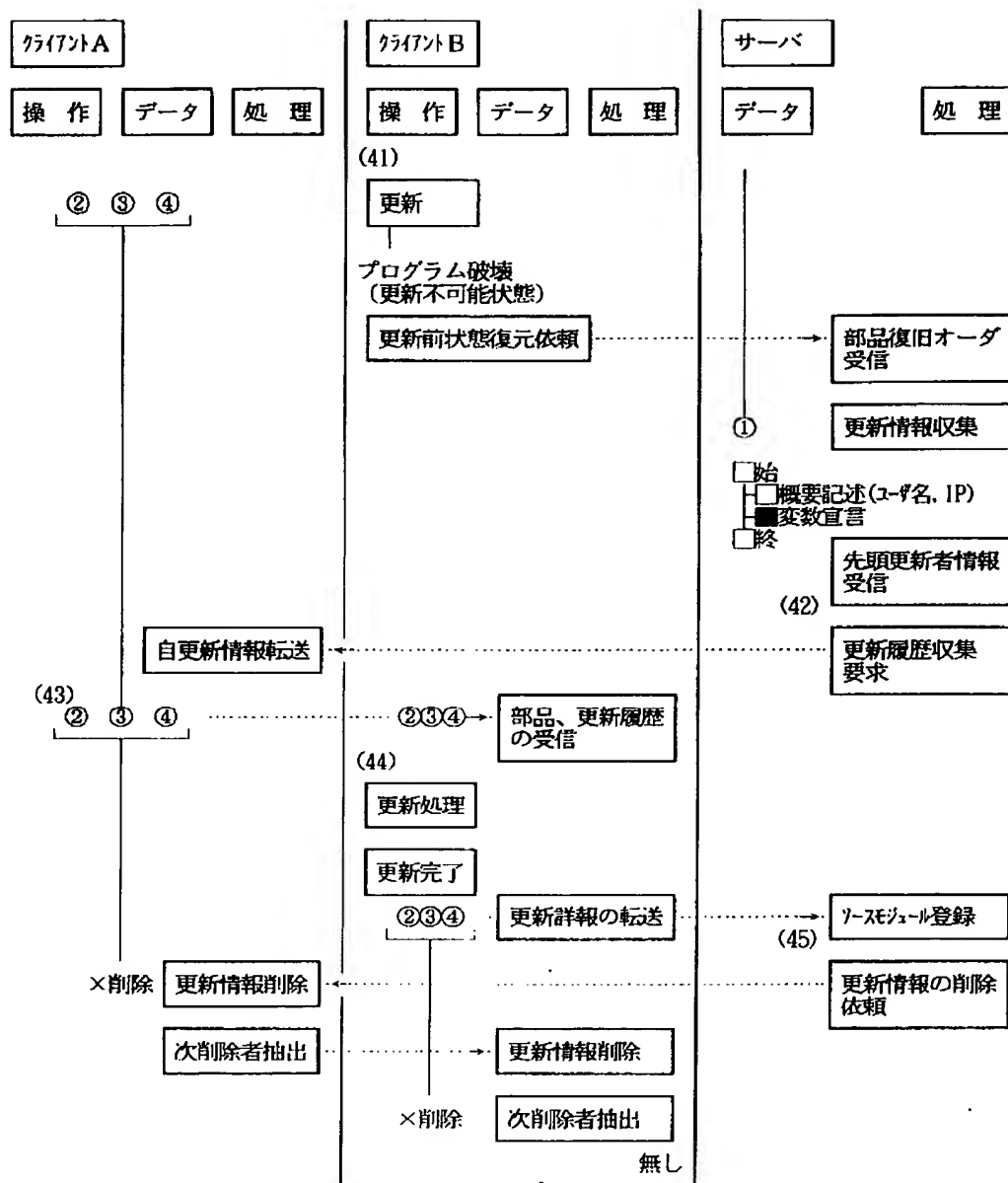
【図9】

実施例5の処理フローチャート（その1）



【図 10】

実施例 5 の処理フローチャート (その 2)



【図 11】

実施例 6 の処理フローチャート

